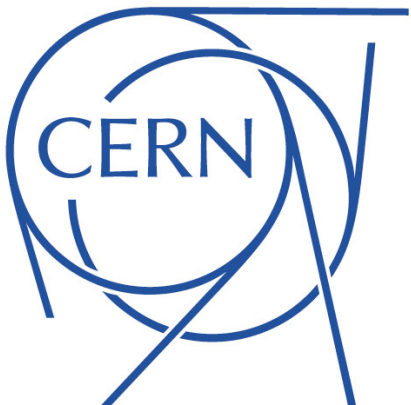


# Geometry & Persistency

## Recent & ongoing developments

Gabriele Cosmo, CERN EP/SFT

*for the Geant4 Geometry & Persistency Working Groups*



# Contents

- Development and fixes in the last year
  - Introduced in release 10.5 and patches
- Features expected in Geant4 10.6
  - Currently under development and scheduled for inclusion in the next release

# Geometry

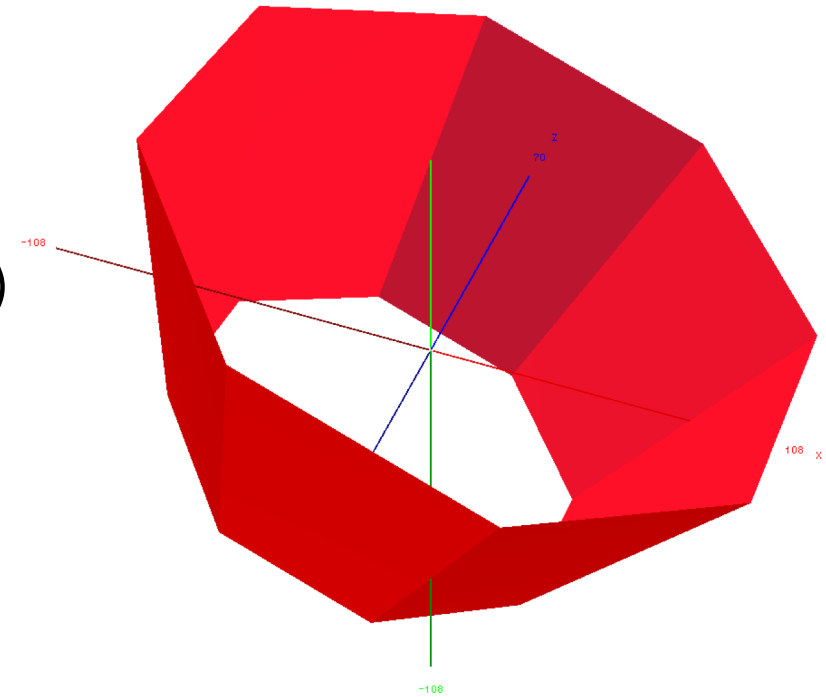
# Solids

- New enabled VecGeom wrappers for *G4ExtrudedSolid*, *G4TessellatedSolid*, *G4Hype* and *G4Tet* 10.5
- New enabled VecGeom wrappers for *G4GenericPolycone*, *G4Ellipsoid*, *G4EllipticalTube* and *G4EllipticalCone* 10.5.ref08
- Improved new version of the VecGeom library 10.5.ref08
  - External library VecGeom v01.01.04
    - <https://gitlab.cern.ch/VecGeom/VecGeom/tree/v01.01.04>
- Reimplemented EstimateSurfaceArea() for approximate calculation of the surface area of a solid 10.5
  - New more performant algorithm, providing more accurate estimation
- Speedup in G4SolidStore, G4Region, G4Tet and G4TessellatedSolid 10.5.p01
  - For initialisation of complex flat geometries
- Revised algorithms and improved robustness in G4EllipticalTube 10.5.p01
- Accurate calculation of radical in G4IntersectingCone for G4Polycone 10.5.p01
  - Problem report [#2111](#)
- Fixed bug in G4ExtrudedSolid::DistanceToOut() 10.5.ref08
  - Affecting extruded solids defined off-center along the z-axis
- Fixed normal from G4Tubs::DistanceToOut() to be unit vector, for abnormal exit points 10.6-β

# VecGeom, release v01.01.04

## *New features since v00.05 series*

- New SIMD helper class “TessellatedSection”
  - Representing a surface made of quadrilateral tiles, delimited by two Z planes, organized in clusters of size = vector length
  - Using explicit vectorization on tiles with VecCore types
  - Can represent sections of multi-faceted solids
    - Polyhedra, extruded solids, trapezoids
    - Improved performance to general extruded-solid and generic-trapezoid
- Added multi-union structure, featuring bounding volume hierarchy (BVH) optimization for vectorized search of candidates
  - Bringing excellent speedup on most queries
- Introduction of new shapes
  - tetrahedron, ellipsoid, elliptical-tube, elliptical-cone
- Removed deprecated USolids module and interfaces
  - Cleanup of build scripts and configuration setup
- Introduced factory for specialized instantiation of shapes
  - Revised most shapes with update of internal API
- Introduced generation of polyhedral meshes for all shapes
- Persistency: first implementation of dedicated GDML reader for VecGeom



# Volumes & Navigation

- Replaced use of Inverse() from G4AffineTransform with new dedicated methods
  - Avoid creation of temporaries in G4RegularNavigation, G4Navigator and G4ReplicaNavigation
- Reviewed geometry workspace classes
  - Make use of generic templated pool class
- Return correct value for multiplicity in G4PVDivision and G4ReplicatedSlice
  - Problem report [#2168](#)
- Re-established original performance in initialisation of the geometry in MT mode
  - For geometries with big faceted shapes
- Improved information in G4Exceptions for particles stuck due to multiple zero steps in G4Navigator
- Correction in G4AssemblyVolume destructor to avoid deletion of internal physical volumes, as cleanup is taken care by G4PhysicalVolumeStore
  - Problem report [#2140](#)
- Fix in G4LogicalVolume::AddDaughter() to avoid propagating pointer to field manager if null
  - Problem report [#2145](#)
- Enable UI command `"/geometry/navigator/push_notify"` which was not setup properly
  - Problem report [#2173](#)
- Revised algorithm for overlaps checking for speed up and improved diagnostics

10.5

10.5

10.6-β

10.6-β

10.6-β

10.5.p01

10.5.p01

10.5.ref07

10.5.ref08

# Magnetic Field

- Correction in G4MagneticField to no longer inherit from G4ElectroMagneticField
  - It is no longer possible to use the equation of motion for a pure magnetic field for the case of a mixed electromagnetic field
- Introduced Bulirsch-Stoer method of integration, an alternative to Runge-Kutta based on the mid-point method
- Separate driver derived from G4VIntegrationDriver
  - Implemented using specialisation of G4IntegrationDriver template class
- Change in G4FieldManager::CreateChordFinder()
  - To cope better with case where field argument is null
- Reviewed treatment of looping particles in transportation in field
  - Only stable particles are killed if they 'loop'
  - Unstable particles are now propagated indefinitely
  - Settings still fully under user control
- Enhanced diagnostics for looping particles
- Corrected transformation to local system in G4QuadrupoleMagField

10.5

10.5

10.5

10.5

10.5

10.5

10.5.p01

# More on field & transportation...

*Also expected in release 10.6*

*(\*) See D.Sorokin talk in Parallel 8B*

- Use of interpolation ability of selected Runge-Kutta methods
  - Substantially reduced number of calls to field evaluation

10.5.ref07

- Revision of interpolation with multiple steppers for each substep

10.6-β

- Revise protocol between transportation and tracking to better cope with particles looping in field

- Review accuracy of boundary crossing in field

➤ Recent ALICE and CMS requirement

*(\*) See J.Apostolakis talk in Parallel 8B*

- General geometry code revision for use of C++11 features

10.5.ref08+



# More from *geometry & transportation workplan*

- Implementation of a prototype navigator based on VecGeom
  - Applicable to placed/static geometries only (\*) See S.Wenzel talk in Parallel 8B
  - First attempt based on 'smart' cloning of navigation histories
  - Evaluation of performance
- Separate safety computation and state from navigator
  - Implement strategy for a light-weight base navigator class not holding navigation state
- Profiling and optimization of multiple navigation
  - Revise design and implementation of multiple navigation and coupled-transportation
- Revision of transportation processes (\*) See M.Asai talk in Parallel 8B
  - Specialized transportation processes for neutral and charged particles

# Persistency

# GDDL module

- Clear auxiliary map information in `G4GDMLReadStructure::Clear()`
  - Problem report [#2064](#)
- Fix in `G4GDMLReadStructure::PhysvolRead()` to allow correct import of recursive assembly structures
  - Addressing problem report [#2141](#)
- Added protection to `G4GDMLParser` for dumping geometry only through the master thread. Added extra protection also in reading
  - Addressing problem report [#2156](#)
- Fixed export of optical surface properties
  - Addressing problem reports [#2142](#) and [#2143](#)
- Fix in `G4GDMLMessenger` to avoid UI commands from being broadcasted to worker threads
- Improved reading of optical properties reader, by allowing reuse of the same `G4MaterialPropertyVector` object for identical properties
- Enable of import/export of assemblies envelopes in GDDL
  - Requirement expressed by LHCb

10.5

10.5.p01

10.5.p01

10.5.p01

10.5.ref08


10.6-β

10.6-β

# Analysis

- Improvements to n-tuple merging in row-wise mode
- Addition of analysis "executive" to provide possibility to choose output type at run-time
- Additional flexibility in resetting/deleting histograms
- Review support for writing same histogram/profile on file several times
- Handling of more files by analysis manager

# Thanks !

<	<b>Fri 27/09</b>	Parallel 8B	>
 Print			
PDF			
Full screen			
Detailed view			
Filter			
09:00	<b>Status of VecGeom navigator integration</b>		<i>Sandro Christian Wenzel</i>
	<i>L102, Jefferson Lab</i>		09:00 - 09:20
	<b>Interpolation for field integration, accuracy and benchmark</b>		<i>Dmitry Sorokin</i>
	<i>L102, Jefferson Lab</i>		09:20 - 09:40
	<b>Accuracy of boundary crossing in field</b>		<i>John Apostolakis</i>
	<i>L102, Jefferson Lab</i>		09:40 - 09:50
	<b>Status of separate Transportation for charged and neutral particles</b>		<i>Makoto Asai</i>
	<i>L102, Jefferson Lab</i>		09:50 - 10:00
10:00	<b>Open issues &amp; discussion</b>		
	<i>L102, Jefferson Lab</i>		10:00 - 10:30